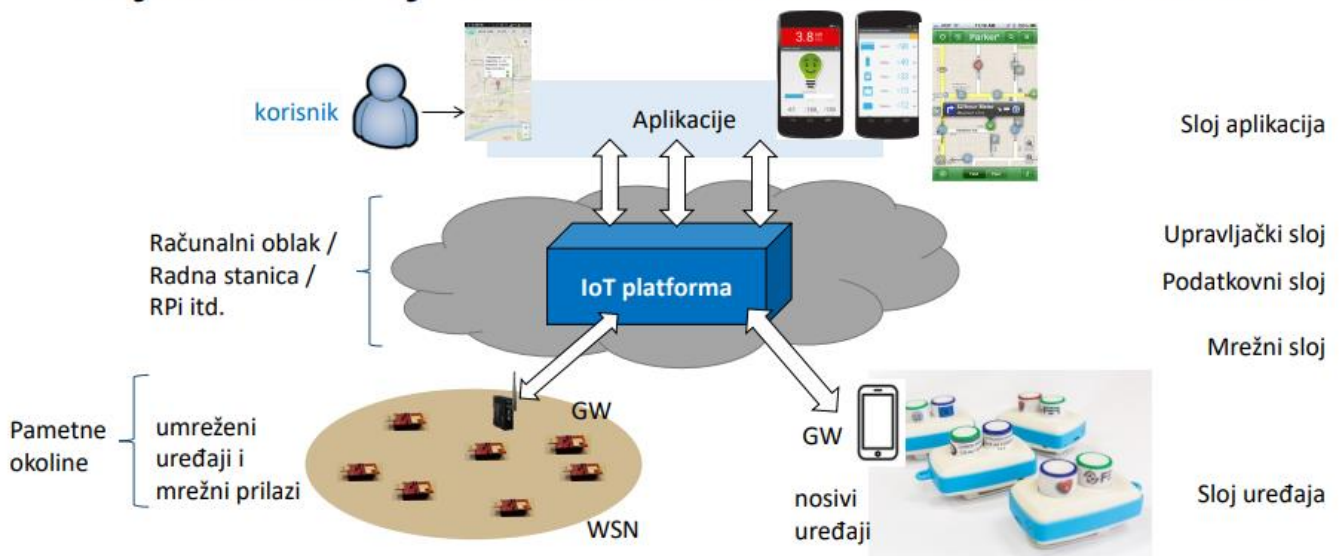


8. IoT-platforme: značajke, komponente, arhitektura, primjeri platformi u računalnom oblaku

Pojednostavljena arhitektura Interneta stvari



IoT-platforma

- Omogućuje povezivanje i upravljanje umreženim uređajima postavljenim u „pametnim okolinama“
- Održava kontinuirano podatke o stanju uređaja
- Prikuplja senzorska očitavanja s umreženih uređaja te omogućuje obradu prikupljenih podataka
- Omogućuje slanje naredbi aktuatorima
- Uređaji postaju dostupni i vidljivi za razvoj korisničkih aplikacija

Tri osnovna skupa funkcionalnosti:

- prikupljanje i pohrana podataka s umreženih uređaja
- upravljanje uređajima i njihovim konekcijama (podržava ograničeni skup protokola, npr. HTTP, MQTT)
- omogućuje razvoj IoT-aplikacija za krajnje korisnike (mobilne i web)

IoT-platforma u računalnom oblaku

Virtualni entitet na nivou računalnog oblaka predstavlja stvarni uređaj

- platforma održava metapodatke o uređajima
- pohranjuje senzorska očitavanja, stanja aktuatora, obrađuje podatke
- vizualizira prikupljene podatke, stanje uređaja, itd.
- mobilne/web aplikacije koriste REST-sučelje platforme za pristup uređajima (platforma je posrednik u toj interakciji)

Računalni oblak

Računalni sustav u kojem se koriste udaljeni poslužitelji putem Interneta, umjesto lokalnih računalnih resursa, za usluge pohrane, obrade i upravljanja podacima

- **Arhitektura:** virtualizirani grozd računala koja se nalaze u jednom ili više podatkovnih centara
- **Svrha** : pružati računalne resurse i usluge na zahtjev
- **Model naplate** : plati onoliko koliko koristiš

Omogućuje fleksibilno korištenje resursa

Mogućnost pokretanja i gašenja virtualnih strojeva po potrebi te rekonfiguracija resursa koji su im dodijeljeni omogućuje **fleksibilnost korištenja**

- frekvencija korisničkih zahtjeva smanjena/povećana -> smanjiti/povećati broj virtualnih strojeva koji ih poslužuju
- rekonfigurirati virtualni stroj te mu dodijeliti više fizičkih resursa

elastičnost: sredstva se mogu po potrebi osigurati i osloboditi, u nekim slučajevima automatski, za brzo skaliranje prema gore i dolje u skladu s opterećenjem

Modeli usluga računalnog oblaka

Software as a Service (SaaS)

- korištenje aplikacija koje se izvode na infrastrukturi oblaka
- GMail, Google Docs, Dropbox

Infrastructure as a Service (IaaS)

- unajmljivanje virtualnih strojeva, zakupljivanje podatkovnog prostora i ostalih računalnih resursa
- DigitalOcean, MS Azure, Google Compute Engine

Platform as a Service (PaaS) - najčešći model IoT-platformi

Nudi se platforma (infrastruktura i razvojna okolina) za umrežavanje uređaja, pohranu podataka i upravljanje uređajima te razvoj korisničkih aplikacija

Korisnik je programer (razvijatelj) IoT-rješenja, prilagođava servise platforme svojim uređajima i području primjene

Korisnici ne kontroliraju OS, bazu podataka, servise platforme, već je samo prilagođavaju svojim potrebama

Brži i jednostavniji razvoj cjelovitih IoT-rješenja, pogodno za manje timove jer se bave konkretnim slučajem uporabe, umrežavanjem uređaja i razvojem korisničkih aplikacija

SW Stack for IoT Cloud Platforms

Connectivity and Message Routing - IoT-platforme trebaju biti u mogućnosti komunicirati s vrlo velikim brojem uređaja i mrežnih prilaza koristeći različite protokole (na sloju podatkovne poveznice i aplikacijskom sloju) i formate podataka, ali zatim je podatke potrebno normalizirati kako bi se omogućila jednostavnija integracija.

Device Registry - središnji registar za identifikaciju uređaja/prilaza koji čine IoT-rješenje.

Device Management - omogućuje upravljanje uređajima, održava podatke o uređajima (metapodaci) i nudi uslugu ažuriranja softvera (firmware).

Data Management and Storage - skalabilna pohrana podataka o uređajima (metapodaci) i s uređaja (telemetrijskih podataka, stanje uređaja) koja podržava količinu i raznolikost IoT-podataka.

Event Management, Analytics & UI - mogućnosti skalabilne obrade događaja, sposobnost analize podataka te izrade izvješća, grafikona i tzv. kontrolnih ploča (dashboard).

Application Enablement - korištenje API-ja za izradu i integraciju korisničkih aplikacija (mobilnih i web) za interakciju s uređajima i vizualizaciju podataka.

Osnovne komponente IoT-platforme

Time-Series Storage & Data Management

- Baza podataka: pohrana podataka, često se podaci sa senzora zapisuju u formatu vremenskih serija (time-series data) – povezuje parametar s vremenskom oznakom

Rule Engine

- Komponenta koja prati poruke na sabirnici poruka i događaje u sustavu te temeljem skupa definiranih pravila izvodi određene akcije
- Može se koristiti i iza „obogaćivanje” poruka

REST API Interface

- Za korisničke aplikacije koje pristupaju senzorskim podacima ili stanju aktuatora
- Korisno za usluge koje trebaju „klasični” pristup podacima zahtjev-odgovor

Device Manager

- Upravlja uređajima, registar uređaja, pregled njihovog stanja (baterija, mreža, vlasnik, lokacija, verzija softvera)

Application and User Management

- Upravljanje korisnicima i njihovim pravima pristupa podacima s uređaja, interakcija i komunikacija s mobilnim aplikacijama, web-aplikacije

Microservices

- Pomoćne usluge, npr. slanje poruka, notifikacija, verifikacija

Amazon AWS IoT

- omogućuje sigurnu, dvosmjernu komunikaciju između IoT-uređaja i AWS oblaka
- nudi niz alata za obradu podataka prikupljenih s umreženih uređaja u računalnom oblaku
- nudi REST API: upravljačko sučelje i provjera stanja spojenih uređaja

Podatkovno sučelje

- dopušta slanje/primanje podataka s uređaja
- Podržani protokoli: MQTT, HTTPS, LoRaWAN

Upravljačko sučelje za Device Shadow(REST)

Sigurnost

Autentifikacija:

- svaki spojeni uređaj mora koristiti certifikat za pristup message brokeru
- Sav promet sa i prema AWS IoT-u mora biti šifriran pomoću protokola Transport Layer Security, **TLS**

Autorizacija

- **autentificirani identitet** će izvršiti AWS IoT operaciju samo ako ima **pravilnik** (policy) koji mu to dopušta
- **Pravilnik** definira što autentificirani identitet (npr. uređaj, mobilna ili web aplikacija) smije napraviti

ThingsBoard

- IoT-platforma otvorenog kôda, mikroservisna arhitektura
- omogućuje upravljanje raznim uređajima i prikupljenim podacima s tih uređaja
- Povezivanje i komunikaciju s krajnjim uređajima: HTTP, MQTT, i CoAP
- Podržava obradu i vizualni prikaz podataka, upravljanje uređajima
- IoT Rule Engine

IoT Data Jedi

- Transformacija, pohrana i usmjeravanje podataka prikupljenih s različitih IoT-uređaja
- Zaštita IoT-uređaja i podataka od neovlaštenog pristupa

9. IoT-platforme i korištenje resursa na rubu mreže

Lokalne platforme za IoT

Upravljanje uređajima seli se iz računalnog oblaka na prilazni uređaj – *gateway*

- tipično uređaji kategorije 2 koji implementiraju puni IP stack
- može se koristiti i poslužitelj u pametnom prostoru
- direktna interakcija aplikacije i platforme u pametnom prostoru (nije putem javnog Interneta)
- „lightweight adapter“

Software stack for GW

Platform Core: minimalni skup funkcionalnosti, ovisi i o tome na kojoj vrsti uređaja se izvodi

Operating System – obično OS opće namjene kao što je Linux

Application Container or Runtime Environment – prilaz često ima mogućnost pokretanja aplikacijskog koda i dinamičkog ažuriranja aplikacija. Na primjer, prilaz može imati podršku za Javu, Python ili Node.js

Communication and Connectivity – prilaz mora podržavati različite protokole za povezivanje s uređajima (npr. Bluetooth, Wi-Fi, Z-Wave, ZigBee, Thread). Prilazi također trebaju biti povezani na Internet (npr. Ethernet, Wi-Fi) i osigurati pouzdanost, sigurnost i povjerljivost komunikacije.

Data Management & Messaging – lokalna pohrana i obrada podataka, mogućnost izvanmrežnog načina rada i obrade u stvarnom vremenu na rubu, postoji i mogućnosti prosljeđivanja podataka prema cloud platformi (podrška za protokole MQTT, CoAP, itd.)

Remote Management – mogućnost daljinskog postavljanja, konfiguriranja, pokretanja/gašenja prilaza kao i aplikacija koje se izvode na prilazima

Home Assistant

- platforma otvorenog koda za upravljanje uređajima u domu, pisan u Pythonu
- praktično korisničko sučelje, moguće definirati situaciju i uvjete izvođenja
- Koristi jezik YAML za konfiguraciju

openHAB

- platforma otvorenog koda za upravljanje domom, predstavlja centralni sustav pametnog prostora
- Jednostavna instalacija na Rpi

Eclipse Kura

EdgeX

- mikroservisi pisani u programskom jeziku Go

Node-RED

- „a visual wiring tool for the Internet of Things“
- Napisan u JS-u, koristi platformu Node.js
- Temelji se na konceptu *Flow-based Programming*

CampusSphere: rješenje za upravljanje uređajima razvijeno u IoT Lab-u

- Koristi više instanci Home Assistant za upravljanje uređajima u pametnim prostorima
- Korisničke role definiraju pravila pristupa i upravljanja uređajima u neposrednoj okolini
- Lociranje korisnika u unutarnjem prostoru: BLE i WiFi
- Tlocrt prostora definiran u formatu Indoor Mapping Data Format

Usporedba lokalnih i cloud platformi

Usporedba prema sljedećem

1. Broj korisnika – skalabilnost
2. Pristup javnom Internetu
3. Generirani promet u javnom Internetu
4. Podržani protokoli
5. Sigurnosni rizici

Ograničenja cloud-platformi

- **Kašnjenje u prijenosu podataka** od uređaja do računalno oblaka i upravljačkih naredbi iz oblaka do uređaja (neke primjene zahtijevaju brzu reakciju)
- Potrebna je **kontinuirana povezanost** na Internet
- **Osjetljivi podaci** ne smiju se pohranjivati na računalni oblak
- Uređaji mogu generirati **velike količine podataka** koje nije moguće ili nema smisla slati u računalni oblak

Računarstvo na rubu mreže

Preraspodjela dijela funkcionalnosti računalnog oblaka na dostupne računalne resurse na rubu mreže

koristi računalne resurse između oblaka i krajnjih IoT-uređaja na putu od uređaja do računalnog oblaka

Obilježja

Heterogena okolina

Nestabilni čvorovi

Česte promjene u mreži

Far edge

zadaci se sele iz računalnog oblaka na mrežne elemente i računalne resurse u blizini IoT uređaja

obrada na rubu mreže: u blizini IoT uređaja, ali ne na IoT uređajima, već jedan skok do njih

umjerena dostupnost računalnih resursa (uređaji ograničenih resursa) uz manju potrošnju energije

omogućuje obradu i prosljeđivanje podataka, upravljanje uređajima te izvođenje pravila uz uvođenje „inteligencije“ i praćenje konteksta na uređajima u blizini IoT-uređaja

prednost: smanjuje vrijeme potrebno za izvođenje usluga i dodatne obrade (ne izvodi se u računalnom oblaku, pa je smanjeno kašnjenje i količina prenesenih podataka u oblak)

Near edge

Koriste se poslužitelji u lokalnoj mreži bliže uređajima, računalni čvorovi s više resursa od onih u far edge - u, npr. lokalni mikro oblak s nekoliko poslužiteljskih rackova

Za računalni oblak na gornjem sloju se podrazumijevaju gotovo neograničeni resursi unutar podatkovnih centara

Prednosti cloud-to-thing computing continuum-a

- veća sigurnost
- veća otpornost na ispade
- smanjeno kašnjenje
- smanjeni operativni troškovi

Svojstva računarstva na rubu mreže - SCALE

Security: Uređaji u edge-u mogu pružiti bolju zaštitu od napada na IoT uređaje ograničenih resursa

Cognition: Odluke se mogu donositi u edge-u umjesto u računalnom oblaku (self-adaptation, self-organization, self-healing...)

Agility: povećava se prilagodljivost pametnog prostora na promjene

Latency: brza reakcija na događanje, smanjuje se vrijeme odziva

Efficiency: poboljšana učinkovitost u odnosu na cloud-rješenja jer se obrada podataka i odluke donose u blizini IoT-uređaja (ali onda mogu biti samo lokalnog karaktera)

Arhitektura - edge

Platform hardware: fizički hardver za edge uređaje

The node management and software backplane: sloj je zadužen za upravljanje čvorovima, omogućuje komunikaciju među svim krajnjim točkama u sustavu (npr. prema udaljenom oblaku, rubnim uređajima, ostalim edge uređajima).

Application support: zbirka mikrousluga koje nisu specifične za aplikaciju. Ovo su generički moduli kao npr. baze podataka, sigurnosni moduli, messaging middleware, etc.

Application Services: usluge namijenjene aplikacijama, za jednostavnu izradu aplikacija koje koriste edge.

Zašto trebamo edge computing?

- Sposobnost okoline da se prilagodi kontekstu

Kontekst je bilo koja informacija pomoću koje se može karakterizirati situacija nekog entiteta.

Entitet je osoba, mjesto ili objekt koji se smatra relevantnim za interakciju između korisnika i okoline

- Okolina je svjesna konteksta ako koristi kontekst kako bi se prilagodila potrebama korisnika u danome trenutku

Vrste konteksta

Korisnički kontekst

- geografska lokacija
- profil korisnika (preference i uzorci ponašanja)

Kontekst uređaja

- stanje senzora i aktuatora (ovisi o vrsti uređaja)
- stanje napajanja
- geografska lokacija

Mrežni kontekst

- Vrsta bežične mreže
- Dostupna širina pojasa, propusnost
- Adresiranje uređaja

Koje se sve obrade mogu izvoditi u edge-u?

Predobrada

- uključuje filtriranje podataka, pronalaženje pogrešnih očitavanja, izdvajanje značajki, transformaciju podataka u prikladniji oblik i dodavanje atributa podacima

Generiranje upozorenja

- kontinuirano praćenje podataka i generiranje upozorenja u slučaju specifičnog očitavanja.
- najjednostavniji primjer je kada temperatura poraste iznad postavljene granice na senzoru.

Joins

- kombinira više tokova podataka u jedan novi tok

Prozor

- Kreira se klizeći prozor nad tokom podatka. Prozori se mogu temeljiti na vremenu (na primjer, jedan sat) ili duljini (2000 očitavanja senzora).
- Na primjer, na temelju 10 posljednjih senzorskih očitavanja računa se srednja vrijednost ili se može brojati značajna promjena temperature u posljednjem satu i utvrditi da će se kvar pojaviti na nekom stroju.

Složeni događaji

- Slijed događaja predstavlja određeni obrazac od interesa koji je potrebno identificirati unutar klizećeg prozora

Strojno učenje

- kreiranje modela koji se kasnije koriste za identificiranje specifičnih stanja analizom dolaznih podataka

Tehnologije za edge

Mikroservisi

kolekcija dobro-definiranih usluga, svaka je neovisna.

- Glavne značajke koje mikroservise čine prikladnima za implementaciju edgea: implementacija neovisna o jeziku
- skalabilnost usluge
- bez centraliziranog upravljanja

Kontejneri

apstrakcija na razini aplikacije koja pakira kod i sve njegove ovisnosti zajedno tako da se može izvršavati jednako i dosljedno na bilo kojoj infrastrukturi.

Kontejnerizacija zahtijeva manje resursa i smanjuje vrijeme pokretanja u usporedbi s virtualizacijom hipervizora.

Svojstva: prenosivost i migracija na različite fizičke čvorove

Alati za orkestraciju kontejnera automatiziraju implementaciju, upravljanje, skaliranje i umrežavanje kontejnera.

Orkestrator je obično centraliziran, koriste se posebni algoritmi za određivanje optimalnog razmještaja servisa na čvorove kako bi se zadovoljili zahtjevi za kvalitetom usluge (npr. kašnjenje)

Federalno učenje - primjer primjene

- Raspodijeljeno udruženo učenje

kognitivni IoT

Kognitivni IoT uvodi kognitivne sposobnosti koje se sastoje od

- ciklusa „percepcije i djelovanja“
- memorije
- pažnje
- inteligencije
- jezika

stvara konvencionalni radni okvir IoT s ciljem omogućavanja interaktivnih usluga

10. Interoperabilnost programskih platformi: programska međuoprema symbloTe i IoT platforma

Što je interoperabilnost?

interoperabilnost je obilježje proizvoda/sustava/programske komponente, čija su sučelja poznata i dobro dokumentirana (tj. otvorena) radi integracije s ostalim sustavima

Vrste interoperabilnosti

Tehnička: povezuje se s komunikacijskim protokolima i potrebnom infrastrukturom kako bi protokoli ispravno funkcionirali.

Sintaktička: povezuje se s formatom podataka i njihovim kodiranjem, npr., XML, JSON, RDF.

Semantička: odnosi se na razumijevanje prenesenih podataka između različitih sustava (informacije, a ne podaci!).

Organizacijska: sposobnost organizacija za učinkovitu komunikaciju i prijenos informacija kroz različite informacijske sustave i infrastrukture

IoT platforma u projektu IoT-polje

Tehnička interoperabilnost

Uređaji

- Waspote Plug & Sense
- ATMOS
- METEOHELIX
- DAVIS
- naši uređaji – STM32, LoRaWAN

Komunikacija:

- **Niži slojevi:** LoRaWAN, Zigbee
- **Aplikacijski:** TLS za sigurnost, MQTT ili AMQP
 - RabbitMQ – message broker

Sintaktička interoperabilnost

- Svaki uređaj šalje podatke u svom formatu
- Ingestion mikro servisi koji parsiraju formate i spremaju u bazu podataka s vremenskim serijama (InfluxDB)

Tehnologije semantičkog weba

Ontologije - Formalni opis domene

Metapodaci

- Podaci o podacima
- Opis objekata pomoću ontologije

Semantičko rasuđivanje - Algoritmi i logika

Ontologija

Eksplicitna i formalna specifikacija

- Formalno se opisuje domena
- Sadrži koncepte (pojmove)
- Definira odnose (predikate) među konceptima

Sastoji se od trojki

- Subjekt
- Predikat / svojstvo / odnos
- Objekt

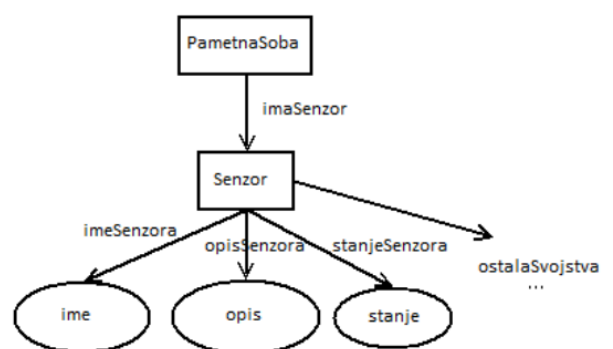
Elementi izgradnje semantičkog weba

- Ontologija i opisna logika - OWL
- Definiranje taksonomija - RDFS
- Povezivanje koncepata - RDF
- Sintakсни sloj - XML(S)
- Protokol WWW-a - HTTP

RDF - Resource Description Framework

Skup W3C-specifikacija koje se danas koriste za **općenito modeliranje informacija**

- Zapisivanje informacija u obliku izjava (trojke): subjekt, predikat i objekt



OWL - Web Ontology Language

- Jezik za definiranje **ontologija**
- Znatno bogatiji vokabular za opisivanje klasa, njihovih međusobnih odnosa, atributa i njihovih svojstava

Jezik SPARQL

- jezik za pretraživanje podataka koji su pohranjeni u RDF-u
- dozvoljava postavljanje nepotpunih upita
- NEDOSTACI:
 - ne posjeduje UPDATE funkciju
 - ne posjeduje kursore
 - nisu mogući upiti računanjem
- SELECT - dohvaćanje u tabličnom formatu
- CONSTRUCT - dohvaćanje u valjanom RDF formatu
- ASK - true ili false odgovori
- DESCRIBE - dohvaćanje u obliku RDF grafova

Primjer jednostavne interoperabilne aplikacije

- Universalni prekidač na pametnom telefonu

symbloTe

IoT portal & Domain Enablers

- **Semantička interoperabilnost**
- registar i semantička tražilica IoT-resursa
- za razvoj servisa s dodanom vrijednosti, pojednostavljuje interakciju s različitim platformama

IoT federations

- **Organizacijska interoperabilnost**
- Zajednice dviju ili više platformi koje dijele ili trguju pristupom svojim IoT resursima
- Aplikacija pristupa resursima svih platformi u federaciji kao da njima upravlja samo jedna platforma
- decentralizirani ekosustav

Core Information Model

Principi dizajna

- što apstraktnije moguće (da ne isključimo druge platforme)
- što eksplicitnije za potrebe (da bi se omogućile sve mogućnosti symbloTe-a)

3 glavne domene

- senzoriranje (temelji se na SSN-u, SensorThingsAPI)
- aktuacija (temelji se na Actua9on-Actuator-Effect, SOSA)
- usluge (motivirano programskim sustavima - poziv procedure)

Iskorištavanje postojećih tehnologija: RDF/S, OWL, W3C Basic Geo Ontology

11. Standardi: standardizacijska tijela i referentne arhitekture. Protokoli za upravljanje uređajima

Standardi

- Tehnički dokumenti dizajnirani da bi se koristili kao pravila, smjernice ili definicije
- **Cilj:** povećati sigurnost proizvoda, smanjiti troškove i cijene

Razlozi za uvođenje standarda

- Mogućnost razmjene informacija, ideja, dobara
- Industrijska revolucija – potreba za tehničkim standardima
- Interoperabilnost nezavisno razvijanih sustava
- Mogućnost suradnje i nadogradnje postojećih sustava

Standardi u računarstvu

- Kodovi za zapis podataka – ASCII, UTF-8
- Hardver – utičnice za spajanje hardverskih komponenti
- Softver – komunikacija između različitih modula, standardizirana sučelja
- TM forum – standardizirana sučelja i informacijski modeli za razvoj interoperabilnih ICT usluga

Internetski standardi

Internet Engineering Task Force

- Zajednica operatora, mrežnih arhitekata, pružatelja usluga i istraživača zadužena za evoluciju internetske arhitekture i rad Interneta

Standardi u IoT-u

- ITU (International Telecommunication Union)
- ETSI (European Telecommunications Standards Institute)
- OMG (Object Management Group)
- W3C (World Wide Web Consortium)
- OMA (Open Mobile Alliance)
- oneM2M

Cilj: uvesti otvorene standarde koji će omogućiti inovaciju i razvoj brojnih novih usluga koje se na te standarde oslanjaju

oneM2M

- Objedinjuje standarde različitih svjetskih organizacija – ETSI, TTA

Dosezi IoT standardizacije

- Standardizirana sučelja za razvoj M2M/IoT aplikacija
- Integracija IoT uređaja i usluga u javnu pokretnu mrežu - **3GPP**
- Specifični slojevi unutar arhitekture IoT platformi
 - usmjeravanje (protokol RPL)
 - IPv6 na uređajima s ograničenim resursima
- Upravljanje uređajima - **OMA**

Referentne arhitekture

oneM2M

Glavni cilj: definiranje međusloja između mreže i aplikacija

Međusloj – IoT platforma

Temeljni dokument: funkcijska (referentna) arhitektura

Tipovi čvorova

- **Logički entiteti**
 - Definirani funkcijama koje posjeduju
 - Podjela: prema pružanju ili nepružanju zajedničkih usluga

Upravitelj sloja aplikacija i usluga

Upravljanje životnim fazama softvera

- Instaliranje
- Ažuriranje
- Deinstaliranje

Različita dostupna stanja softvera

- Idle
- Starting
- Active
- Stopping

Upravitelj podacima i skladištenjem

- Prikupljanje
- Konverzija
- Analiza
- Semantička obrada

Podaci koje su prikupili aplikacijski entiteti

Metapodaci za održavanje sustava:

- Podaci o uređajima
- Dozvole pristupa
- Informacije o pretplatama i lokacijama

Upravitelj uređajima

- Konfiguracija
- Dijagnostika
- Praćenje rada
- Upravljanje softverom
- Upravljanje topologijom
 - Za mrežne prilaze

AIOTI - Alliance for Internet of Things Innovation

Ostale standardizacijske aktivnosti

Upravljanje uređajima

Za **komunikaciju između upravljačkog poslužitelja i upravljačkih klijenata oneM2M** pretpostavlja korištenje postojećih **protokola** za upravljanje uređajima:

- Lightweight M2M (LWM2M)
- OMA Device Management (OMA DM)

Protokoli za upravljanje uređajima

Mogu se koristiti za različite funkcije zajedničkih usluga

- Konfiguracija uređaja
- Ažuriranje softvera
- Nadzor uređaja
- Sigurnost komunikacije između uređaja

Standardizacijsko tijelo OMA

Otvoreni standardi u javnoj pokretnoj mreži

Članovi:

- proizvođači hardvera
- operatori pokretnih mreža
- proizvođači softvera

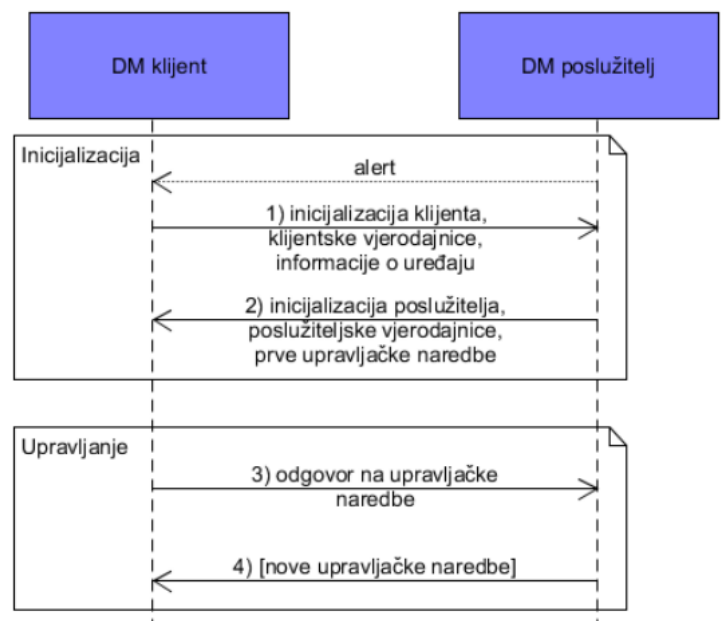
Namjena: upravljanje pokretnim uređajima

Obilježja:

- Oslanja se na protokol HTTP
- DM poslužitelj i DM klijent

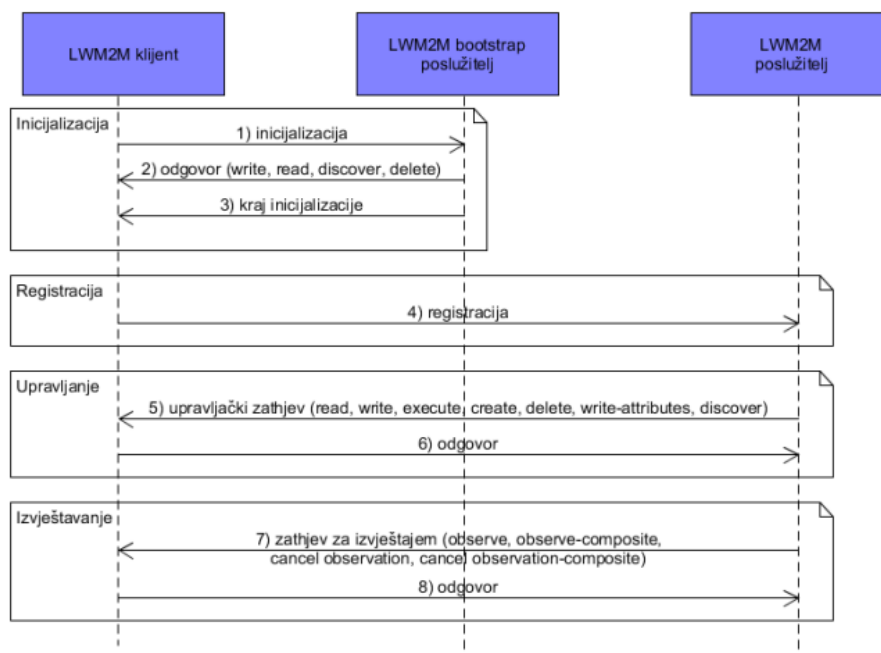
Sigurnost

- autentifikacija i autorizacija



Lightweight M2M (LWM2M)

- Noviji standard organizacije OMA
- Nasljednik OMA DM-a
 - Prilagođeniji M2M/IoT uređajima
 - Izvođenje na uređajima s ograničenim resursima
- **Obilježja**
 - Oslanja se na protokol CoAP
 - LWM2M poslužitelj i LWM2M klijent
- **Sigurnost**
 - autentifikacija i autorizacija



Implementacije

OMA-DM

- Open5GMTC - OMA DM
- Friendly OMA-DM embedded client

LWM2M

- Leshan (Java)
- Wakaama (C)

Standardi organizacijskih tijela

Glavne prednosti

- Rigorozan dizajn
- Responzivnost industrije
- Testiranje i certifikacija

IoT standardizacija - poteškoće

- Duplicirane IoT-arhitekture i modeli
- Veliki broj komunikacijskih protokola za heterogene IoT uređaje
- Podatkovni modeli su vlasnički i razvijaju se za specifične, vertikalne domene
- Nedostaje usklađenosti procesa obrade podataka nastalih senzorskim mjerenjima
- Sigurnost i privatnost se razmatraju na pojedinačnim slojevima
- Lako korištenje i održavanje IoT rješenja zahtjeva globalni pristup

12. Aplikacije Interneta stvari: stvarnovremenske usluge, pametni grad, pametni dom i ured

Područja primjene

- Smart home
- Smart city
- Poljoprivreda
- Logistika
- Zdravlje

Različiti zahtjevi za primjere usluga

| | Pametna brojila (potrošnja el. energije) | Intelligent Transport System | Nadzor (kamere, senzori za prisutnost i sl.) | |
|-------------------------------|--|------------------------------|--|------------------|
| | Smart meters | eHealth | ITS | Surveillance |
| Mobility | none | Pedestrian /vehicular | Vehicular | none |
| Message size | Small (few kB) | Medium? | Medium | large |
| Traffic pattern | Regular | Regular/irregular | Regular/irregular | Regular |
| Device density | Very high (up to ten thousands per cell) | Medium | High | low |
| Latency requirements | low (up to hours) | Medium (seconds) | Very high (few milliseconds) | Medium (< 200ms) |
| Power efficiency requirements | High (battery powered meters) | High (battery power devices) | Low | low |
| Reliability | High | High | High | medium |
| Security requirements | High | Very high | Very high | medium |

Važni zahtjevi o kojima treba voditi računa prilikom pokretanja projekta:

- **mobilnost** - da/ne?, uređaj na osobi, uređaj na vozilu
- **veličina poruke** - mala/srednja/velika
- **obrazac slanja poruka** - redovito/neredovito
- **gustoća mreže uređaja** - niska/srednja/visoka/jako visoka
- **tolerancija kašnjenja**
- **efikasnost napajanja**
- **pouzdanost**
- **sigurnosti zahtjevi**

Pametni dom

- Ambijentalne usluge
 - grijanje i hlađenje
 - upravljanje osvjetljenjem
 - nadzor prisustva u domu
 - upravljanje kućanskim uređajima
 - lociranje u zatvorenim prostorima
- Ušteda energije
 - pametna brojila
 - upravljanje kućanskim uređajima ovisno o cijeni energije
- Usluge za starije osobe i osobe s posebnim potrebama
 - puls
 - kisik u krvi
 - disanje
 - tjelesna temperatura
 - položaj tijela

Koncept pametnog grada

Umreženi uređaji i inovativne IKT-usluge su pokretači razvoja održivih i tehnološki naprednih gradova u službi građana i gradskih službi.

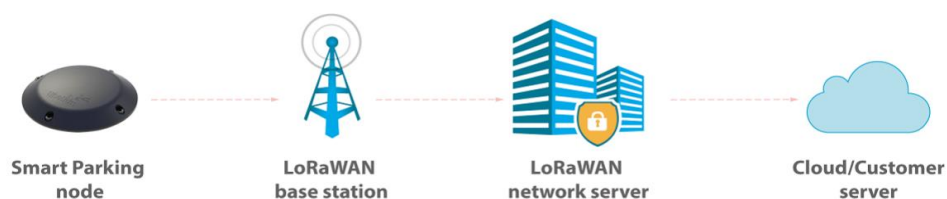
Promet – nadzor prometnica, javni prijevoz, parking

Zbrinjavanje otpada

Ambijentalni nadzor – razina buke, kvaliteta zraka

Energija – pametna rasvjeta

Primjer rješenja za pametno parkiranje



Ostale domene primjene

Tvornice

- praćenje proizvodnog procesa - temperatura, tlak
- praćenje radnih uvjeta - osvjetljenje, temperatura, buka, vlaga
- reguliranje emisije štetnih plinova

Praćenje poplava

Praćenje nuklearnog zračenja

Nadzor u poljoprivredi

Stvarnovremenske usluge

Današnje IoT-aplikacije: praćenje stanja okoliša i nadzor različitih sustava

- kontinuirano se prikupljaju velike količine podataka i provodi analitika nad podacima
- podaci su u formi tokova podataka (time-series data, niz podataka indeksiranih vremenskim oznakama)

Dodatni zahtjevi: mogućnost upravljanja (aktuacije)

- **Stvarnovremenost:** odluku o akciji je potrebno donijeti u kratkom vremenskom intervalu („u stvarnom vremenu“) nakon čega kreće **izvedba upravljanja** (aktuacija)
- Otpornost na gubitke podataka
- Zadovoljavanje performansi upravljanja

Stvarnovremenski problem

Postoji upravljački tok podataka F_i koji utječe na provedbu aktuacije

- Senzor $\rightarrow \dots \rightarrow$ upravljačka jedinica $\rightarrow \dots \rightarrow$ aktuator
- Generiranje paketa u periodu P_i
- Postoji više kontrolnih petlji u mreži senzora, uređaja i aktuatora

Svaki tok podataka F_i mora biti obrađen u roku $D_i \leq P_i$ (stvarnovremenski zahtjev)

- Stabilnost i predvidivi upravljački nadzor

Izazovi

- Zadovoljavanje krajnjeg roka za provedbu aktuacije
- Mogućnost planiranja vremena izvršavanja

Stvarnovremenski zahtjevi

Near real-time

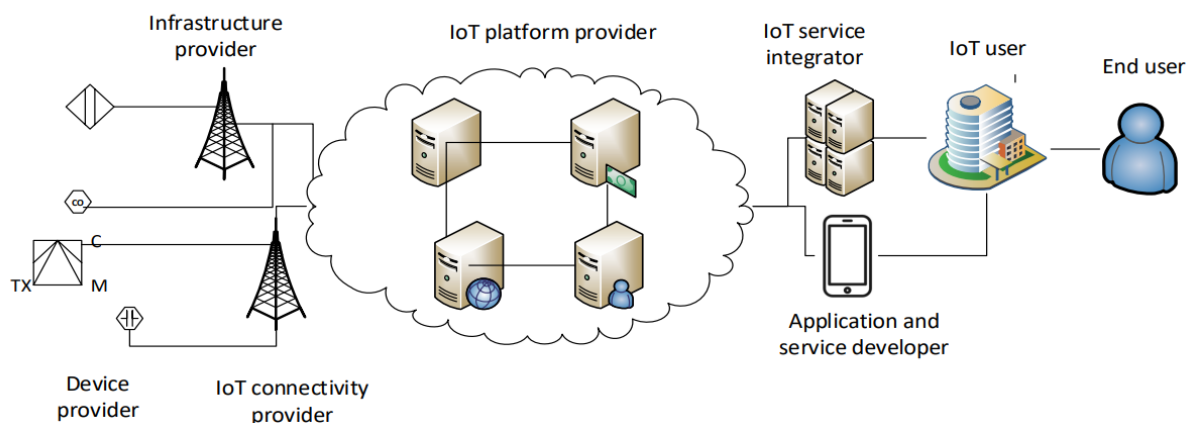
- Tolerancija kašnjenja od nekoliko sekundi do nekoliko minuta

Sustavi objavi - pretplati

- Koliko je vremena potrebno od objave do isporuke obavijesti svim pretplatnicima?
- Za koliko sekundi treba pokrenuti akciju ako je opažen određeni događaj ili niz događaja?

Zahtjevi za sustav

- **Hard** - propuštanje roka isporuke uzrokuje ispad sustava
- **Firm** - propuštanje roka isporuke se tolerira, ali degradira kvalitetu usluge; rezultat nema značaj nakon roka isporuke
- **Soft** - propuštanje roka isporuke se tolerira, ali degradira kvalitetu usluge



Infrastructure provider: pruža IoT-uređaje i infrastrukturu koja osigurava povezanost uređaja na Internet.

- **Device provider:** proizvođač IoT-uređaja
- **IoT connectivity provider:** mrežni operator

IoT platform provider:

- nudi IoT-platfomu za jednostavan pristup senzorskim podacima i upravljanim uređajima kao i integraciju podataka za razvoj novih usluga.
- Odgovoran je za kontrolu heterogenih uređaja

IoT service integrator:

- integrator nad IoT-platfomom i uređajima
- nudi usluge u određenoj domeni primjene te dodaje vrijednost servisima IoT-platfome
- Koristi po potrebi usluge razvoja mobilnih i web-aplikacija

IoT user:

- kupac IoT-usluge koji koristi nove umrežene komponente u svojim proizvodima (npr. pametno brojilo)
- pruža inovativne usluge (npr. uslugu pametnog mjerenja) krajnjim korisnicima.

End user: krajnji korisnik IoT-usluge

Kako pristupiti dizajnu novog IoT-rješenja?**Razmisliti o tehničkim karakteristikama:**

- 1. Osnovne karakteristike**
 - a. broj krajnjih korisnika
 - b. količina podataka
 - c. vremenska osjetljivost
 - d. lokacija
 - e. financijski model
 - f. skalabilnost
- 2. Karakteristike uređaja**
 - a. način komunikacije
 - b. računalne mogućnosti
 - c. potrošnja energije
 - d. izvor energije
 - e. lokacija uređaja
- 3. Karakteristike komunikacije/povezivosti**
 - a. one way/two way
 - b. širina pojasa komunikacije
 - c. tolerancija kašnjenja
 - d. tolerancija gubitka podataka
- 4. Karakteristike platforme**
 - a. lokalna/globalna
 - b. edge uređaji / gateway uređaji
 - c. podržani protokoli
 - d. servisi za analizu podataka

Strateški cilj projekta AloTwin

značajno **ojačati znanstvenu izvrsnost i inovacijski kapacitet** FER-a u području Interneta stvari kroz prijenos znanja i suradnju s vrhunskim znanstvenicima iz vodećih europskih institucija

stvoriti poticajno istraživačko okruženje na FER-u i povećati znanstvenu kvalitetu i produktivnost kako bi privukli nove nacionalne i međunarodne talente

ojačati kapacitete institucije za sudjelovanje u projektima **Obzor Europa**